NUMERICAL RADIATIVE TRANSFER USING A MULTIGRID METHOD

Master of Science in Astronomy Thesis

Johan Pires Bjørgen
Institute of Theoretical Astrophysics
University of Oslo
Norway

June 2014

Abstract

Solving the non-LTE radiative transfer problem in stellar atmospheres is computationally demanding. There is therefore an interest to investigate new methods to solve this problem effectively. A numerical method, multigrid, is a promising candidate to achieve this goal. It was originally developed to solve boundary problems in the 60s, but recently the method has been applied to several different problems. The important advantages of the multigrid scheme are the very high convergence rate, and that the convergence speed does not deteriorate when the discretization is refined. Both properties are highly desirable for solving the radiative transfer problem. I have implemented a non-linear multigrid scheme around an existing radiative transfer code, RH. This multigrid scheme achieves high convergence speed on very fine-grids, at most eight times faster than without multigrid. The results show that a non-linear multigrid method can handle various atmospheres and atoms.

Key words: radiative transfer, solar atmosphere, multigrid

Acknowledgments

First I would like to thank Jorrit Leenaarts, which provided a interesting project with a focus on a numerical method, Multigrid, and a really difficult topic. I appreciate the countless Skype meetings we had through the completion of this thesis. And thanks too Mats Carlsson, with whom I had a lot of discussion about how to modify the radiative code. I am grateful to ITA (Viggo Hansteen), which sent me to Stockholm to work a whole week with Jorrit, which resulted in a working code with multigrid! Thanks to IAESTE Oslo/Norway to keep me occupied with other things than working at my thesis at in spare time and study times. And thanks to T-banen to let me drive trains philosophizing on my master thesis while earning money at the same time and for giving me vacation in May/June for working with the thesis. Without the students in the basement it would have been a boring year, without the chess and ping-pong matches. And finally I would like to thanks the creators of Tinder, which helped to "meet" my girl during the thesis work by making it possible to work in the basement and flirt at the same time (It's not only in science new discovers is done due to better methods, but also in dating!)

And special thanks to my spell-checking army, without them this thesis would be impossible to read:
Jorrit Leenaarts
Mats Carlsson
Martin Rødvand
Vedad Hodzic
Benedicte Emilie Brækken

Contents

1. Introduction **9**

2. Radiation transfer **11**
2.1. Radiation . . . 11
2.1.1. Basic quantities . . . 11
2.2. Numerical . . . 14
2.2.1. Non-LTE . . . 14

3. Multigrid **17**
3.1. Background . . . 17
3.2. Iterative scheme . . . 18
3.2.1. Jacobi method . . . 19
3.2.2. Gauss-seidel . . . 20
3.3. Grids . . . 21
3.4. Operators . . . 21
3.4.1. Interpolation . . . 21
3.4.2. Restriction operator . . . 22
3.5. Linear multigrid . . . 23
3.5.1. The Multigrid Cycle . . . 26
3.5.2. W-cycle . . . 27
3.6. Non-linear multigrid . . . 27
3.6.1. Example: Heat conduction . . . 29

4. Radiative transfer: Multigrid **33**
4.1. About the RH code . . . 33
4.2. Implementation . . . 34
4.2.1. Algorithm . . . 34
4.2.2. Equation system . . . 35
4.2.3. Particle Conservation . . . 35
4.2.4. Atom . . . 35
4.2.5. Atmosphere . . . 35
4.3. Analyzing . . . 38
4.3.1. Error . . . 38
4.3.2. Convergence . . . 38
4.3.3. Convergence criteria . . . 39

4.3.4. Post/Pre-smoothing details 40
4.4. Results 40
4.4.1. Smoothing numbers 40
4.4.2. Number of coarse grids 45
4.4.3. Hydrogen 45
4.4.4. Restriction 45

5. Conclusion and outlook **49**
5.1. Improvements 49

Appendices **51**

A. Definition **53**

Bibliography **55**

1. Introduction

In astrophysics we are studying objects which are far from us, in most cases we can't experiment on site. So most of the information we get from these objects depend on other type of information carriers. For example we can use radar, landers, orbiters etc., to gather information from the objects we want to study. But the main wealth of diagnostic information is provided by electromagnetic radiation. All objects observable to us, which are not obscured by other objects, emit EM waves, photons. These travel at the speed of light and do not decay on the way. This provides us rich information from the objects we want to study. They can tell us about the local conditions from which the photons originate, direction, energy level, wavelength, etc. These quantities are important to understanding what type of conditions and processes are on the object, or undergoing. Having proper diagnostic tools to analyze and understand the information is critical, this links the observations with theory. One important branch of astrophysics which addresses these problems, is radiative transfer. Most realistic cases for solving radiative transfer, are very complex. Often we tend to use numerical models to look at these problems.

Carrying out a numerical analysis is essential to understanding the problem to be solved, with the computational tools we have available and finally construct an algorithm for solving the given problem to a desired accuracy, within the limits of computational cost and time. This thesis will address these problems. It aims to implement a non-linear multigrid with an existing radiative transfer code, RH, for solving the radiative transfer problem. This is a step to investigate if this works for realistic cases, and provide a framework which can be utilized in the future.

The layout is as follow as: Chapter 2 will cover a short introduction of radiative transfer and explain why this is hard to solve, Chapter 3 we have a look at the theory behind the multigrid and understand how this can be used as an advantage for solving a radiative transfer problem. Finally in Chapter 4 we show the implementation of multigrid in a radiative transfer code, RH, and and present our results. At the end, in Chapter 5 a conclusion and outlook are given.

2. Radiation transfer

Most of the observations we do of the Sun are done in the electromagnetic spectrum, which is used to study the Sun in great detail. The observations mainly come from the atmosphere, which can be divided into different layers. One of them is the photosphere which is a little more than hundred of kilometers in thickness and have a temperature 5770 K [12]. This is where the gas goes from transparent to almost opaque, so most of the information we receive, comes from this layer. Where and how deep the photon originates, depends on the wavelength, and the position we look at the Sun. Which means that we can study the Sun in great detail using observations. Linking the simulations with the observation, is in great interest to understand more of the properties of the Sun.

In this chapter we will go trough how the definition of quantities to understand how the energy is transferred, and how we can calculate this properties.

2.1. Radiation

We will go through the foundation of radiative transfer which is needed to understand what happens to a beam of light when it passes trough a system. The following definition and formalism is based on Rutten [8, chap. 2-3].

2.1.1. Basic quantities

Intensity

The intensity, I_ν, is defined as:

$$I_\nu = \frac{dE_\nu}{\cos\theta dA dt d\nu d\Omega} \tag{2.1}$$

where dE_ν is the amount of energy transported through the surface dA, at the location $\vec{r}$ and with $\vec{n}$ normal to dA, between a small timescale dt time and a frequency band, $d\nu$ and in the solid angle $d\Omega$ in direction $\vec{l}$, see Figure 2.1. It has the units [erg cm^{-2} s^{-1} sr^{-1}Hz^{-1}]. The index denotes the frequency, ν. The intensity along a beam is constant unless there are processes which add or remove photons from the beam.

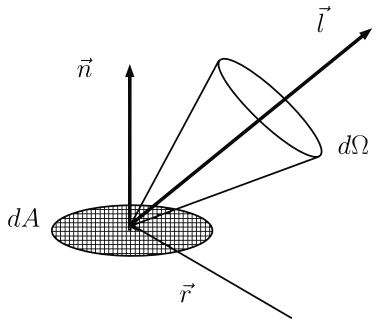


Figure 2.1.: Illustration of amount of energy transported through the surface dA, at the location $\vec{r}$ and with $\vec{n}$ normal to dA, between a small timescale dt time and a frequency band, $d\nu$ and in the solid angle $d\Omega$ in direction $\vec{l}$

Mean intensity

The mean intensity is averaging intensity, $I_\nu(\theta, \phi)$ over all the directions, is defined

$$J_\nu \equiv \frac{1}{4\pi} \int I_\nu d\Omega = \frac{1}{4\pi} \int_0^{2\pi} \int_0^{\pi} I_\nu \sin\theta d\theta d\phi \tag{2.2}$$

The units are the same as for the intensity, I_ν. This quantity is interesting when we want to know only the availability of the photons and not the direction.

Emissivity

The monochromatic emissivity , j_ν, per cm^{-3} is defined by:

$$j_\nu = \frac{dE_\nu}{dV dt d\nu d\Omega} \tag{2.3}$$

where the $\mathrm{d}E_\nu$ is the energy locally added to the radiation. The dimension of j_ν is [erg $\mathrm{cm}^{-3}\mathrm{s}^{-1}$ Hz^{-1}]

The emissivity shows the amount of intensity added by photon emission to a beam:

$$dI_\nu = j_\nu(s) ds \tag{2.4}$$

Extinction

The monochromatic extinction, α_ν , specifies the energy fraction taken from a beam

$$dI_\nu = -\alpha_\nu I_\nu ds \tag{2.5}$$

which has the dimension, cm^{-1}.

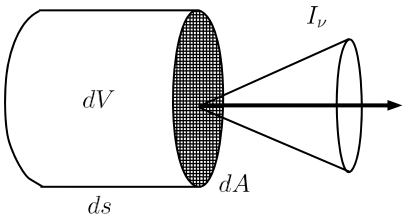


Figure 2.2.: Illustration of a beam with a cross-section dA passes trough a cylinder with a volume dV.

Transport equation

With Eq. 2.4 and 2.5, we can defined the total change of the intensity of a ray passing trough a medium:

$$dI_\nu = -\alpha_\nu I_\nu ds + j_\nu ds \tag{2.6}$$

which is often called the transport equation. This sums up the effects of the emission and extinction to the variation of the intensity passing a length, ds, see Figure 2.2.

The photons do not care about the length, but rather how tick the medium is, we define a quantity which is measured along the beam that takes this in account:

$$d\tau_\nu = \alpha_\nu ds \tag{2.7}$$

Which is the optical path, integrating the total optical path of a medium with the length D, we get the optical thickness:

$$\tau_\nu(D) = \int_0^D \alpha_\nu(s) ds \tag{2.8}$$

which can be described as the penetration of the photon in the medium.

Another important quantity, which describe the ratio between emission and the extinction coefficient:

$$S_\nu = \frac{j_\nu}{\alpha_\nu} \tag{2.9}$$

which is called the source function and has the same units as intensity.

Using this the transport equation can be written as the change of the optical thickness

$$\frac{dI_\nu}{d\tau_\nu} = S_\nu - I_\nu \tag{2.10}$$

which tells us the variation of the intensity with the optical depth. This can be integrated, along the beam in a medium with a length $s = 0$ to $s = D$:

$$I_\nu(D) = I_\nu(0)e^{-\tau_\nu(D)} + \int_0^{\tau_\nu(D)} S_\nu(s)e^{-(\tau_\nu(D)-\tau_\nu(s))}d\tau_\nu(s) \tag{2.11}$$

which is called the formal solution.

2.2. Numerical

In this section, I will show which equations are needed to be solved in a radiative transfer problem. This is a small overview.

2.2.1. Non-LTE

Non-LTE stands for non-local thermodynamical equilibrium, which is used when the assumption for local-thermodynamic equilibrium fails. Then we often assume statistical equilibrium, which is one of the three equations we need to solve for in the radiative transfer problem.

The statistical equilibrium is expressed as:

$$n_i \sum_{j\neq i}^{N} P_{ij} - \sum_{j\neq i}^{N} n_j P_{ji} = 0 \tag{2.12}$$

there n_i is the population number for a certain level i and P_{ij} is the probability for a atom in level i making a transition to level j, and N is the number of levels.

Particle conservation is the second equation

$$\sum_{j=1}^{N} n_j = n_{tot} \tag{2.13}$$

and third one is transport equation:

$$\frac{dI_\nu}{ds} = -\alpha_\nu I_{\nu\mu} + j_\nu \tag{2.14}$$

This problem is non-local and nonlinear which makes it hard to solve. To see this we need to study the equations closer. Equation 2.12 can be written out with the probabilities P_{ij} as

$$P_{ij} = R_{ij} + C_{ij} \tag{2.15}$$

where the R_{ij} is the contribution from the radiative part, and C_{ij} is the collisional contribution. It can be further written out the radiative contributions for bound-bound transition as:

$$R_{ij} = A_{ij} + B_{ij}\overline{J}_{ij}, \quad i > j \tag{2.16}$$
$$= B_{ij}\overline{J}_{ij}, \quad i < j \tag{2.17}$$

where A_{ij} and B_{ij} are the Einstein coefficient for spontaneous emission probability: stimulated emission ($i > j$) and absorption ($i < j$). Similar equations hold for the bound-free equations. $\overline{J}_{ij}$ depends on the emissivity and extinction from all the points in the whole atmosphere, and those quantities depends on the particle densities. Therefore Eq. 2.12 is non-linear and non-local.

This makes the radiative transfer a hard problem to solve.

3. Multigrid

There are many approaches of studying a physical problem often, apply a numerical method to, since it is impossible to solve analytically or is tedious work. There are many approaches and several methods to do this, depending on the problem to be solved. Multigrid is a class of numerical methods to solve PDE and other types of equations. The main idea behind multigrid, is using different grids to acquire the solution to the problem. The advantage with this method is how the problem is resolved on coarser grids, fewer grid points, and the computationally cost to solve the problem is less. This chapter will cover the fundamental components for every multigrid algorithm: iterations methods, sequences of grids, a transfer operator between grids, a relaxation operator (smoothing scheme), a coarse grid version of the fine grid operator and a coarse grid operator solver. Here the coarse grid operator solve can just be the relaxation operator, letting it converge to a certain point or a direct solver. The linear and non-linear, where for the non-linear scheme we will only cover the fast approximation scheme, both schemes will be utilized on a test example to see how it works.

For readers being more curious of the mathematics behind multigrid, a good converge analysis of linear and non-linear multigrid can be found in "Multi-Grid methods and Applications" by Hackbush [7]. For readers wanting to get a quick overview over all the different methods and analysis with less theory, "A Multigrid Tutorial" by William L. Briggs [14], is recommended.

3.1. Background

Multigrid is often referred to as multilevel methods, first developed in 1961 by R.P Fedorenko, a Soviet mathematician, who proposed the two-grid method in his paper Fedorenko [2] for solving a two dimensional Poisson's equation with iterative methods. Later he did a convergence analysis Fedorenko [3] which led to the whole multigrid method and laid the foundation for the main scheme of the method. In the 70s the method was reinvented independently by A. Brandt and W. Hackbush. A good historical overview of how the multigrid has developed the last 50 years can be found in "Multigrid methods: from geometrical to algebraic versions", [6].

Today multigrid has been applied to a multitude of different problems, such as algebraic problems, image reconstruction, optimization, statistical mechanics, quantum chromodynamics and integral equations. The method is also highly

customisable, for example, it can be used with non-uniform or variable-mesh problems.

3.2. Iterative scheme

This section will set up the notation being used throughout the thesis. Having a set of linear equations, and a right hand side, it can expressed as

$$A\mathbf{u} = \mathbf{f} \tag{3.1}$$

Suppose the system has a unique solution $\mathbf{u}$ for the Eq. 3.1, and trying to solve it with an approximation $\mathbf{v}$ which is close to the unique solution. The error of the approximation can then be written as

$$\mathbf{e} = \mathbf{u} - \mathbf{v} \tag{3.2}$$

where $\mathbf{v}$ is the approximation to the solution and $\mathbf{e}$ is the error. This tells us how close we are to the solution. Calculating the error is as hard as finding the actual solution to the problem. If the error is not known, we can compute the residual instead:

$$\mathbf{r} = \mathbf{f} - A\mathbf{v} \tag{3.3}$$

This equation is a measure of how well $\mathbf{v}$ approximates the solution $\mathbf{u}$. The residual does not tell us how close the approximation is to the solution, so if the residual is small this does not mean the error is small as well. The residual is zero if and only if the given approximation, $\mathbf{v}$, is the given solution $\mathbf{u}$ to the problem.

Combining the two Eq. 3.2 and 3.3, one can derive an equation for the error, this is called a residual equation:

$$\mathbf{r} = \mathbf{f} - A\mathbf{v} \tag{3.4}$$
$$\mathbf{r} = A\mathbf{u} - A(\mathbf{u} - \mathbf{e}) \tag{3.5}$$
$$A\mathbf{e} = \mathbf{r} \tag{3.6}$$

Solving this equation, the error is found, then the approximation can be updated and the solution can be acquired:

$$\mathbf{u} = \mathbf{v} + \mathbf{e} \tag{3.7}$$

Solving Eq. 3.6 is as computationally demanding as solving Eq. 3.1. If the error is smooth, Eq. 3.6 it can be solved on a coarse grid. But how is the approximation found in the first place? Typically when solving a physical problem, for example radiative transfer, the initial guess for the radiation field is set to LTE values,

then applying an iteration scheme will lead us closer to the solution, hence better approximation. In the following sections I will discuss two iteration schemes, the Jacobi and Gauss-Seidel iteration schemes.

3.2.1. Jacobi method

Wanting to solve a problem with the approximation,$\mathbf{v}$,in this form

$$A\mathbf{v} = \mathbf{f}, \tag{3.8}$$

the matrix A can be written out as

$$A = L + D + U \tag{3.9}$$

Here L is the strictly lower and U strictly upper triangular part of the matrix A, and D is the diagonal matrix. Inserting this into Eq. 3.8 leads to this system

$$(L + D + U)\,\mathbf{v} = \mathbf{f} \tag{3.10}$$
$$D\mathbf{v} = \mathbf{f} - (L + U)\,\mathbf{v} \tag{3.11}$$
$$\mathbf{v}^{m+1} = D^{-1}\mathbf{f} - D^{-1}\left((L + U)\,\mathbf{v}^{m}\right) \tag{3.12}$$
$$\mathbf{v}^{m+1} = D^{-1}\mathbf{f} - R_J\mathbf{v}^{m} \tag{3.13}$$

where the $R_J = D^{-1}(L + U)$ is the Jacobi iteration matrix, m is how many iterations done on the approximation. The whole system can be written in element wise as

$$v_i^{m+1} = \frac{1}{a_{ii}}\left(f_i - \sum_{j\neq i}^{n} a_{ij}v_j^m\right) \tag{3.14}$$

To see why the updated approximation is converging to the solution, it is necessary to look at how the error changes after each iteration, and see if the error is reduced.

Analysis of convergence

Converging to the wanted solution is done by updating the approximation. Looking at how the error changes for each iteration provide insight to when it does converge, using the definition from Eq. 3.7. If the exact solution is used to iterate, then there should not be any changes when applying an iteration, $\mathbf{u} = R\mathbf{u} + g$, where $\mathbf{g} = D^{-1}\mathbf{f}$. Using the definition it can be rewritten as

$$\mathbf{e}^{m+1} = \mathbf{u} - \mathbf{v}^{m+1} \tag{3.15}$$
$$\mathbf{e}^{m+1} = R\left(\mathbf{u} - \mathbf{v}^{m}\right) + \mathbf{g} - \mathbf{g} \tag{3.16}$$
$$\mathbf{e}^{m+1} = R\mathbf{e}^{m} \tag{3.17}$$

telling us how much the error changes with one sweep with the iteration matrix, R. This can be defined generally for m iterations with the initial guess

$$\mathbf{e}^{m} = R^{m}\mathbf{e}^{0} \tag{3.18}$$

It can be shown that the scheme will converge

$$\lim_{m\to\infty} R^{m} = 0 \tag{3.19}$$

If and only if the $\rho(R) < 1$, called the convergence factor of the iteration matrix. The ρ is the spectral radius of the matrix, which is the largest eigenvalue of the matrix, R.

3.2.2. Gauss-seidel

The Gauss-Seidel is an iteration scheme quite similar to Jacobi's iteration scheme, with a minor change. The new computed approximation, v^{m+1} are used as soon as they are computed, so essentially:

$$v_i^{m} \rightarrow v_i^{m+1} \tag{3.20}$$

Here the arrow denotes replacement. Looking again at Eq. 3.8 and writing it out with Eq. 3.9 the Gauss-Seidel iteration matrix can be derived:

$$(L + D + U)\,\mathbf{v} = \mathbf{f} \tag{3.21}$$
$$(D + L)\mathbf{v} = \mathbf{f} - (U)\,\mathbf{v} \tag{3.22}$$
$$\mathbf{v} = -\left(D + L\right)^{-1} U\mathbf{v} + \left(D + L\right)^{-1}\mathbf{f} \tag{3.23}$$
$$\mathbf{v}^{m} \leftarrow \left(D + L\right)^{-1}\mathbf{f} - R_G\mathbf{v}^{m} + \tag{3.24}$$

It can be shown that Gauss-Seidel has a better convergence and smoothing capabilities than the Jacobi's iteration scheme. But it is more complicated, since $(D + L)$, has to be inverted. The whole system can be written in element form as

$$v_i^{m+1} = \frac{1}{a_{ii}}\left(f_i - \sum_{j<i}^{n} a_{ij}v_j^{m+1} - \sum_{j>i}^{n} a_{ij}v_j^{m}\right) \tag{3.25}$$

3.3. Grids

In multigrid equations will be solved on different grids. The finest grid will be denoted as Ω^h, where h gives the spacing between the points, $h = \frac{l}{n}$, where l is the length of the grid and n is the number of points. So the coarser grid essential have less points, making the grid spacing become larger:

$$\Omega^h > \Omega^{2h} > \Omega^{4h} ... \tag{3.26}$$

A coarse grid will have just half the number of points (#) from the finer grid:

$$\frac{1}{2}\#\Omega^h = \#\Omega^{2h} \tag{3.27}$$

Usually there is no advantage in choosing a higher or lower ratio for the grid spacing in multigrid.

3.4. Operators

To transfer a quantity from a coarse-grid to a fine-grid and vice versa, a mechanism is needed. Two methods will be discussed here, interpolation and restriction.

3.4.1. Interpolation

Many interpolation methods could be used, but for most cases a linear interpolation is enough in multigrid schemes. Interpolating from a coarser grid, Ω^{2h}, to a finer grid, Ω^h, will be denoted by I^h_{2h}.

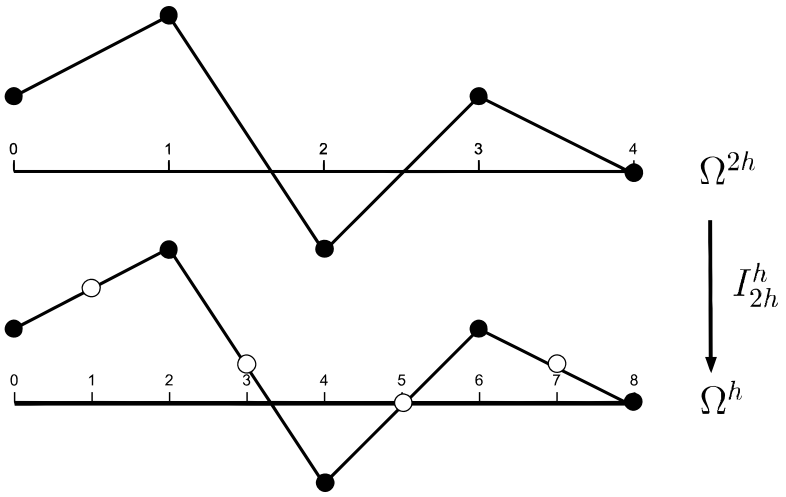


Figure 3.1.: Example of interpolation of a vector from a coarser grid Ω^{2h} with 5 points to a fine grid Ω^h with 9 points.

A linear interpolation is given by

$$y = \quad y_0 + (y_1 - y_0)\left(\frac{x-x_0}{x_1-x_0}\right) \tag{3.28}$$

In this multigrid scheme uniform grids will be used, and the interpolation will be a straight average:

$$y = \quad \frac{y_0+y_1}{2} \tag{3.29}$$

Numerically our interpolation operator, I_{2h}^h, is given by

$$v_{2j}^h = v_j^{2h} \tag{3.30}$$

$$v_{2j+1}^h = \frac{1}{2}\left(v_j^{2h} + v_{j+1}^{2h}\right) \quad , 0 \leq j \leq \frac{n}{2} - 1 \tag{3.31}$$

Here the j is the index of the coarser grid spacing. Figure 3.1 shows graphically the operator I_{2h}^h.

3.4.2. Restriction operator

Restriction from a finer grid, Ω^h, to a coarser grid, Ω^{2h}, will be denoted by R_h^{2h}. Two methods will be used to, injection and weighting. The easier one is using injection, meaning taking every second point on the fine grid and transfer it to the coarse grid, see Eq. 3.32, and as seen in Figure 3.2. Regarding computational work, this is the most optimal one.

$$v_j^{2h} = v_{2j}^h \tag{3.32}$$

Doing injection, we ignore information from the odd points, so to remove any problems with injection that may arise, full weighting comes in. There we average the neighbor points at the finest grid before restricting the point to the coarser point:

$$v_j^{2h} = \frac{1}{4}\left(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h\right) \tag{3.33}$$

This is a three-point evaluation, where these weights have been applied $\{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$. One can use other weights to regulate the influence from the neighboring points.

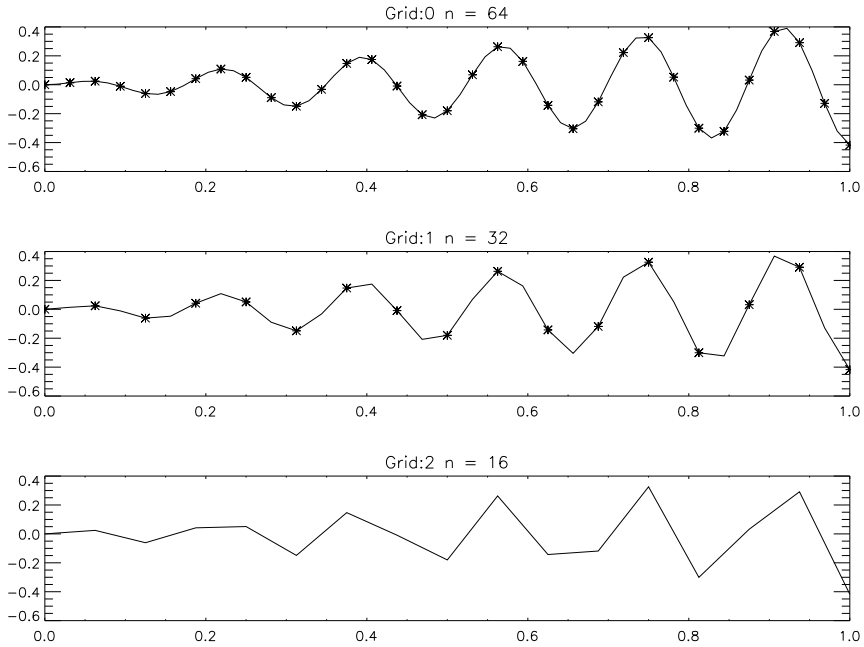


Figure 3.2.: Shows a wave restricted with the injection method, Eq. 3.32, to coarser levels. The upper panel represented the wave on a grid with 64 points, the star symbols represent the points which is restricted to the middle panel, which consist of 32 points. At the middle panel marks again with star symbols which is restricted to the lower panel, which is the coarsest grid with 16 points.

3.5. Linear multigrid

Referring to Section 3.2, where the definition of the linear problem we wanted to solve was on the form, $A\mathbf{v} = \mathbf{f}$. Computing the residual of the system, $\mathbf{r} = \mathbf{f} - A\mathbf{v}$, the residual equation can be used to find the exact error:

$$A\mathbf{e} = \mathbf{r} \tag{3.34}$$

If the error and residual are both smooth, smoother than the solution, $\mathbf{u}$, one can represent Eq. 3.34 on a coarser grid. Here smooth means that the high-frequency components of the error and residual are small or zero. The iterations schemes Jacobi and Gauss-Seidel remove these high-frequency components, but not the low-frequency components of the error.

The big advantage of solving the problem on a coarser grid is based on two ideas: less points on the coarser grid, requiring less computational work need to be

done and the smoothing capabilities is more effective here. Using the restriction operator, it can therefore represent the residual at the coarser grid:

$$\mathbf{r}^{2h} = R_h^{2h}\mathbf{r}^h \tag{3.35}$$

the new linear system is

$$A^{2h}\mathbf{e}^{2h} = \mathbf{r}^{2h} \tag{3.36}$$

which can be solved exact for the error with less work. If using a relaxation scheme to solve this problem, an initial guess set to $\mathbf{e}^{2h} = 0$ is often good enough. We can also apply a direct solver to solve this system, hence solving the system exact. After the exact error or a good estimation of the error has been obtained on the coarser grid, it can be transferred back to the finest grid using the interpolation operator. We can now update our current approximation

$$\mathbf{v}_{new}^h \leftarrow \mathbf{v}_{old}^h + I_{2h}^h\mathbf{e}^{2h} \tag{3.37}$$

which should be closer to the solution of the system. Since the error has been solved at a coarser grid with less information then the original system we intended to solve, high-frequency interpolation noise has been introduced. As we can not represent the system exact at a coarser grid with fewer points, this is inevitable. Applying some iterations (for example Jacobi method) will get rid of this noise, this is called post-smoothing. This algorithm is called the two-grid cycle.

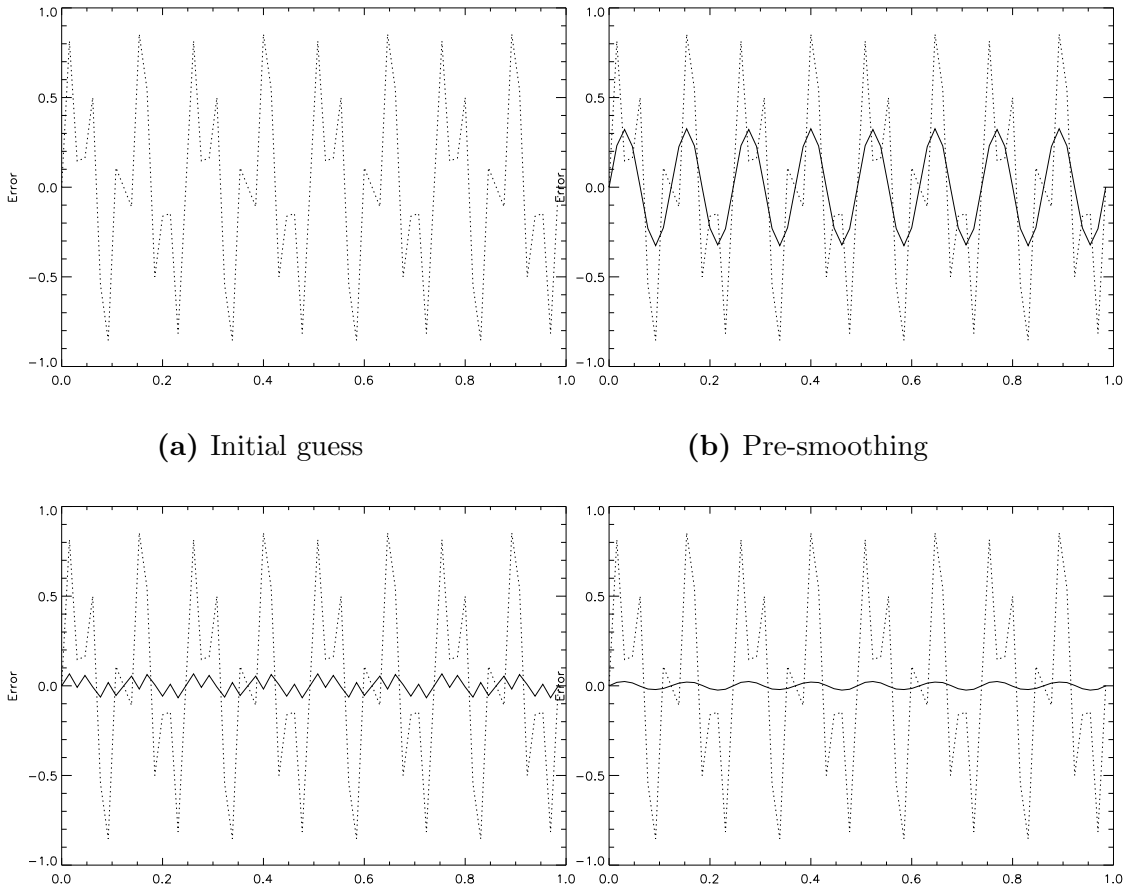
The scheme is:

1. Perform ν_1 pre-smoothing
2. Compute $\mathbf{r}^h = \mathbf{f}^h - A^h\mathbf{v}^h$
3. Restrict the residual to a coarser grid, $\mathbf{r}^{2h} = R_h^{2h}\mathbf{r}^h$
4. Solve the equation $A^{2h}\mathbf{e}^{2h} = \mathbf{r}^{2h}$
5. Interpolate the correction to a finer grid, $\mathbf{v^h} \leftarrow \mathbf{v^h} + I_{2h}^h\mathbf{e}^{2h}$
6. Perform ν_2 post-smoothing

Where ν is how many iterating (smoothing) operations used.

Example: Two-points boundary problem

The result of the scheme can be seen in Figure, 3.3, where the two-grid scheme was applied to a two-point boundary value problem, mainly the Poisson equation, $-u''(x) = 0$, with boundary condition $u(0) = 0$ and $u(1) = 1$. The solution to this problem is, $\mathbf{u} = 0$. This problem could be solved with a much easier scheme, but the purpose here is to see how well the idea of multigrid works.



(a) Initial guess **(b)** Pre-smoothing

(c) Coarse-grid-correction **(d)** Post-smoothing

Figure 3.3.: The error (solid line) of the problem, $u''(x)$, with the initial guess (dotted lines) seen in Eq. 3.38.

The initial guess, seen in Figure 3.3a, consists of two modes with different wave-numbers,

$$\mathbf{v} = \frac{1}{2}\left(\sin\left(16\pi\mathbf{x}\right) + \sin\left(40\pi\mathbf{x}\right)\right) \tag{3.38}$$

Applying one sweep with pre-smoothing, one can see that the high-frequency error has been removed in Figure 3.3b and we are left with the low-frequency error. To effectively remove it, we transfer the residual to Ω^{2h}, a coarser grid. Here the low-frequency is represented in a much higher frequency band, and therefore we apply two coarse-grid-correction finding the error. Having computed the error we transfer the correction back to the fine grid with a linear interpolation to update the approximation. Now we introduced an extra error, interpolation noise, as seen in Figure 3.3c. The interpolation noise can be removed by using two post-smoothing in Figure 3.3d.

3.5.1. The Multigrid Cycle

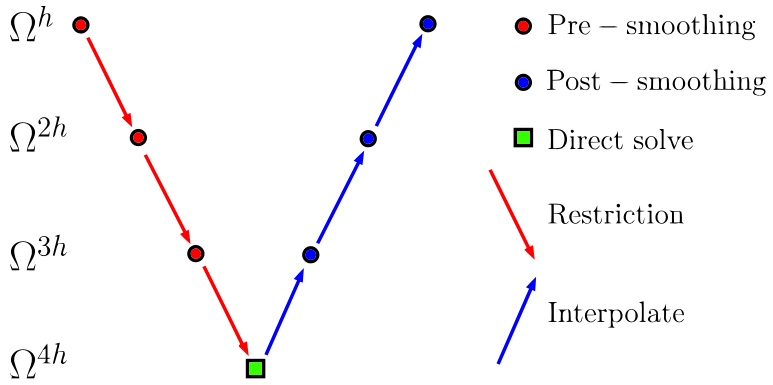


Figure 3.4.: General V-cycle, which show the problem is restricted with $n_l = 4$

In the two-grid method, the coarse grid equation

$$A^{2h}\mathbf{e}^{2h} = \mathbf{r}^{2h} \tag{3.39}$$

is of the same form as the original problem we wanted to solve, $A\mathbf{v} = \mathbf{f}$ (Eq. 3.1). We can therefore apply the same two-grid techniques on the coarse grid equation, which is called the multigrid method. The multigrid scheme where one performs consecutive coarsening followed by interpolations to the finest grid is called a V-cycle, see Figure 3.4 for a graphical representation. The scheme for a four-grid multigrid can be written like this:

- Perform ν_1 pre-smoothing $A^h\mathbf{v}^h = \mathbf{f}^h$
- Compute $\mathbf{f}^{2h} = R_h^{2h}\left(\mathbf{f}^h - A^h\mathbf{v}^h\right)$
  - Perform ν_1 pre-smoothing, $A^{2h}\mathbf{v}^{2h} = \mathbf{f}^{2h}$
  - Compute $\mathbf{f}^{4h} = R_{2h}^{4h}\left(\mathbf{f}^{2h} - A^{2h}\mathbf{v}^{2h}\right)$
    - Perform ν_1 pre-smoothing, $A^{4h}\mathbf{v}^{4h} = \mathbf{f}^{4h}$
    - Compute $\mathbf{f}^{8h} = R_{4h}^{8h}\left(\mathbf{f}^{4h} - A^{4h}\mathbf{v}^{4h}\right)$
      - Solve, $A^{8h}\mathbf{v}^{8h} = \mathbf{f}^{8h}$
    - Interpolate , $\mathbf{v}^{4h} \leftarrow \mathbf{v}^{4h} + I_{8h}^{4h}\mathbf{v}^{\mathbf{8h}}$
    - Perform ν_2 post-smoothing
  - Interpolate , $\mathbf{v}^{2h} \leftarrow \mathbf{v}^{2h} + I_{4h}^{2h}\mathbf{v}^{\mathbf{4h}}$
  - Perform ν_2 post-smoothing
- Interpolate , $\mathbf{v}^h \leftarrow \mathbf{v}^{4h} + I_{2h}^h\mathbf{v}^{2h}$
- Perform ν_2 post-smoothing

This is the full power of linear multigrid cycle. The coarsest grid which can be chosen in a ordinary multigrid cycle is three points, one interior point and two boundary points. It is not certain if the problem being solved can be represented on such a coarse grid. The number of grids being used is uncertain and require adjustment and experimentation.

3.5.2. W-cycle

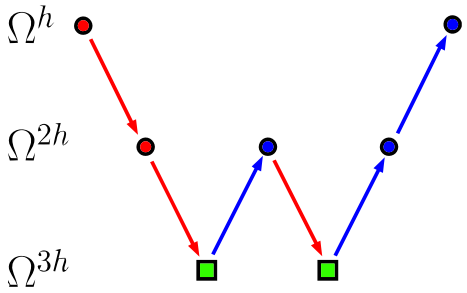


Figure 3.5.: General W-cycle, which show the problem is restricted with three grids. It does two coarse-grid-correction, instead of one as in the V-cycle.

There exists different type of cycles, and one of them is called W-cycle, as seen in Figure 3.5. For some applications the W-cycle may be faster than the V-cycle. In this thesis only the V-cycle is applied.

3.6. Non-linear multigrid

Multigrid can be applied on non-linear systems as well, but because of the non-linearity one need a different approach than for the linear case. There exist several methods for applying non-linear multigrid, but in this thesis only the full approximation scheme (FAS) is used.

$$A(\mathbf{u}) = \mathbf{f} \tag{3.40}$$

here the non-linear operator, A, is working on $\mathbf{u}$. The approximation to the solution $\mathbf{v}$ and the error is $\mathbf{e} = \mathbf{u} - \mathbf{v}$. The residual is $\mathbf{r} = \mathbf{f} - A(\mathbf{v})$. Using these two relations, the residual can be expressed as:

$$\mathbf{r} = A(\mathbf{u}) - A(\mathbf{v}) \tag{3.41}$$

which is the same as for the linear case. But since the operator is non-linear, one can not derive a simple equation for the error:

$$A(\mathbf{e}) \neq A(\mathbf{u}) - A(\mathbf{v}) \tag{3.42}$$

Meaning we need a different approach to represent the problem on a coarser grid. Using the idea from Section 3.5, we can represent Eq. 3.41 on a coarse grid, Ω^{2h}

$$\mathbf{r}^{2h} = A^{2h}(\mathbf{u}^{2h}) - A^{2h}(\mathbf{v}^{2h}) \tag{3.43}$$

The solution of the coarse-grid is written as

$$\mathbf{u}^{2h} = \mathbf{v}^{2h} + \mathbf{e}^{2h} \tag{3.44}$$

where the $\mathbf{v}^{2h}$ and $\mathbf{e}^{2h}$ is the coarse-grid approximation from the finer grid, $\mathbf{v}^h$ and $\mathbf{e}^h$. The residual can be represented with a restriction operator, thus Eq. 3.43 can be written as

$$\mathbf{f}^{2h} = A^{2h}(\mathbf{v}^h) + R_h^{2h}\mathbf{r}^h \tag{3.45}$$

Here a difference arise from the linear scheme, a coarse-grid approximation, v^{2h}. Restricting our approximation from the finest grid, the right-hand-side from Eq. 3.45 is known, and therefore the right-hand-side be defined as:

$$\mathbf{f}^{2h} = A^{2h}(I_h^{2h}\mathbf{v}^h) + R_h^{2h}\mathbf{r}^h \tag{3.46}$$

The equation we want to solve on the coarse grid:

$$A^{2h}\left(\mathbf{u}^{2h}\right) = \mathbf{f}^{2h} \tag{3.47}$$

Instead of solving for the error as in the linear case, we solve for an approximation on the coarse grid, with a modified right-hand-side. When the solution for the coarse-grid is acquired, we calculate the error, $\mathbf{e}^{2h} = \mathbf{u}^{2h} - \mathbf{v}^{2h}$ and update our approximation on the fine grid:

$$\mathbf{v}_{new}^h \leftarrow \mathbf{v}_{old}^h + I_{2h}^h\mathbf{e}^{2h} \tag{3.48}$$

which should be closer to the solution of the system. Afterward we apply some post-smoothing to remove the high-frequency components from the interpolation.

For a non-linear two-grid can be describe like this:

1. Perform ν_1 pre-smoothing
2. Restrict the residual and current approximation $\mathbf{r}^{2h} = R_h^{2h}\left(\mathbf{f} - A^h(\mathbf{v}^h\right)$ and $\mathbf{v}^{2h} = I_h^{2h}\mathbf{v}^h$
3. Solve the residual equation, $A^h(\mathbf{u}^{2h}) = A^{2h}(\mathbf{v}^{2h}) + \mathbf{r}^{2h}$

4. Compute the coarse-grid approximation to the error $\mathbf{e}^{2h} = \mathbf{u}^{2h} - \mathbf{v}^{2h}$

5. Interpolate the error up to the fine-grid and correct the approximation $\mathbf{v}^h_{new} \leftarrow \mathbf{v}^h_{old} + I^h_{2h}\mathbf{e}^{2h}$

6. Perform ν_2 pre-smoothing

This scheme is called Fast Approximation Scheme since we solve for an approximation instead of the actually error for the problem. This method could be applied for several grids, with the same idea as in Section 3.5.1.

3.6.1. Example: Heat conduction

The heat conduction equation is used as an example which can be solved with the FAS multigrid algorithm. The Gauss-Seidel scheme is used as the smoothing operator.

The equation for heat conduction can be written as

$$\frac{\partial T}{\partial t} = \kappa_0 \frac{\partial}{\partial x}\left(T^{5/2}\frac{\partial T}{\partial x}\right) \tag{3.49}$$

where the T is the temperature, and κ_0 is the thermal conductivity. To simplify the problem we look at the steady-state problem, we set the time derivative to zero and $\kappa_0 = 1$,:

$$0 = \frac{\partial}{\partial x}\left(T^{5/2}\frac{\partial T}{\partial x}\right) \tag{3.50}$$

which is a second-order nonlinear differential equation, and can also be written as

$$0 = \frac{5}{2}T^{3/2}\left(\frac{dT}{dx}\right)^2 + T^{5/2}\frac{d^2T}{dx^2} \tag{3.51}$$

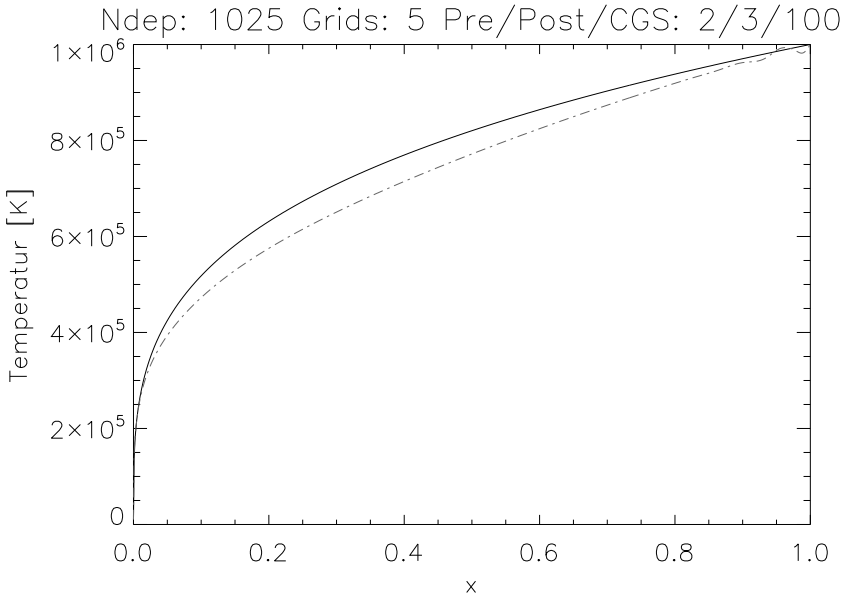
this equation is discretized as follows,

$$0 = \frac{5}{2}T_i^{3/2}\left(\frac{T_{i+1} - T_{i-1}}{2\Delta x}\right)^2 + T_i^{5/2}\left(\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2}\right) \tag{3.52}$$

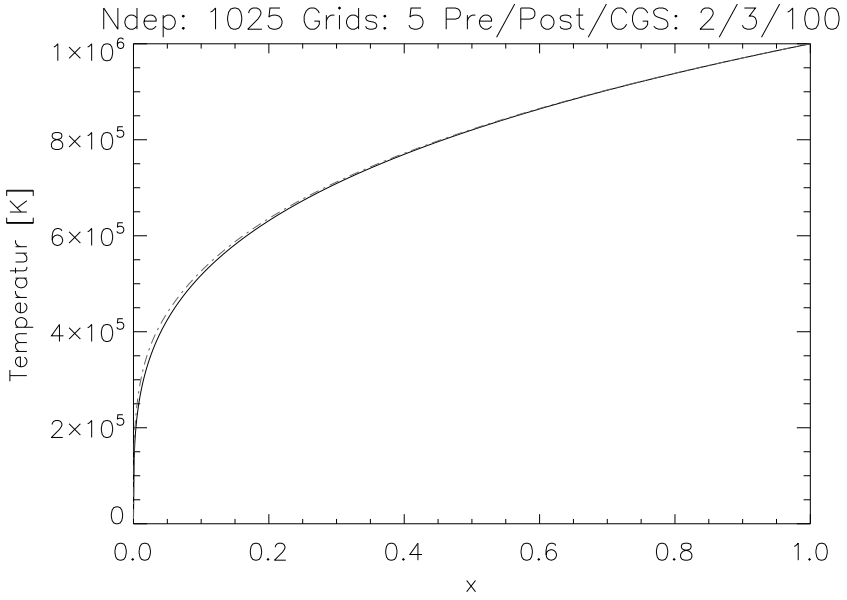
This equation is solved using the following boundary conditions

$$T(0) = 10^4 K \tag{3.53}$$
$$T(1) = 10^6 K \tag{3.54}$$

We solve this problems with Gauss-Seidel and FAS.



(a) Multigrid method has a relative maximum error below 0.10, comparable with 112 pure iterations Gauss-Seidel



(b) Converged solution

Figure 3.6.: Temperature variation across the domain. The solid line is the exact solution to heat conduction, the dotted/dashed-line is the solution for the non-linear multigrid.

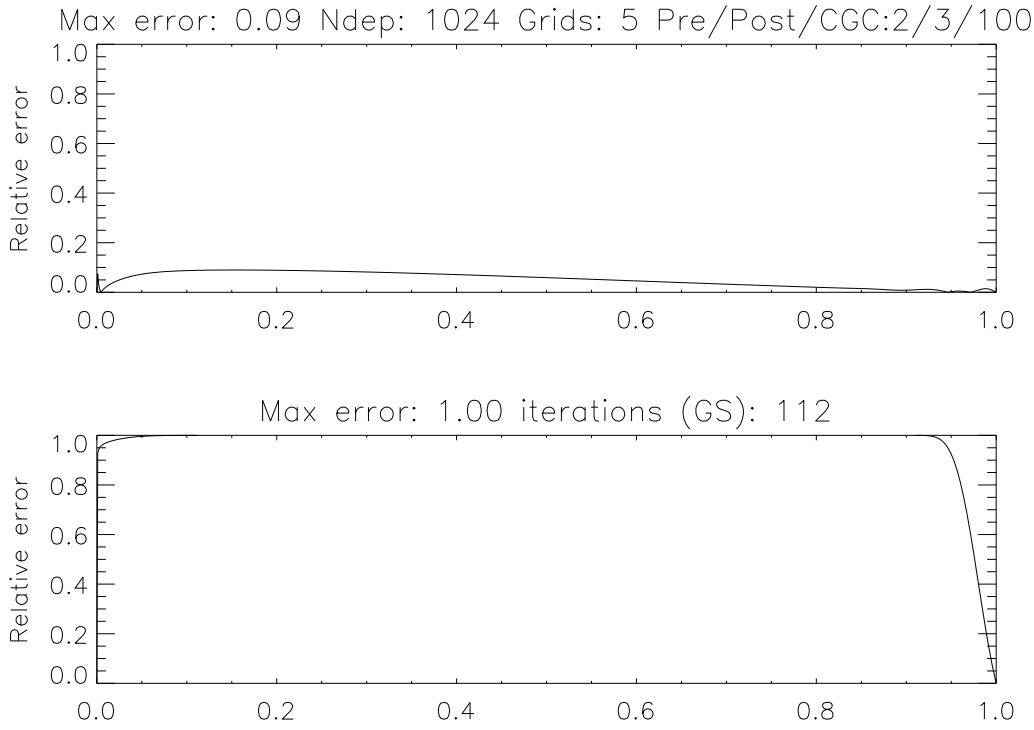


Figure 3.7.: The relative error across the domain with 1024 points. The upper panel shows the multigrid with five grids against the lower panel which is pure Gauss-Seidel iterations. It has been done 112 GS iterations at both cases.

Results

The solution to the heat conduction is analytically known, making it possible to assess whether the non-linear multigrid has converged to the right solution. The solution with the multigrid solution is shown in Figure 3.6, where the Figure 3.6a has converged where the maximum relative error is below 0.10. In Figure 3.6b is the converged solution, where it can be seen it has problem to converge to the exact solution, at the temperature gradient. This problem however was also apparent in the pure Gauss-Seidel iterations, and are not limited to multigrid.

A closer look at the performance of the multigrid is shown in Table 3.1, where the overhead of the administration part of the multigrid is not taken in account. Already at 128 points, the speed-up is significant, but advantage of multigrid becomes apparent at 512 points, where multigrid is 64 times faster.

Points	Gauss-seidel [itr]	Multigrid (GS) [itr]	Speed up
128	3046	141	21 $\times$
512	49185	160	307 $\times$
1024	198593	112	1773 $\times$

Table 3.1.: Comparison with non-linear Gauss-Seidel against FAS. With the multigrid method we applied $\nu_1 = 2$, $\nu_2 = 3$, 100 coarse-grid-correction with the grids 2,4,5 respectably for points 128,512,1024.

4. Radiative transfer: Multigrid

Today there exist fast methods for solving non-LTE multilevel problems in 1D using operator splitting methods with high convergence speed. In these cases we do not need a higher convergence speed, since the computational time needed is short. If we want to solve complicated non-LTE transfer problem in 3D combined with detailed physics and a fine grid, we encounter problems. With today's methods, the convergence speed goes down when the number of grid points are increased in the atmosphere, which is the cases in 3D. The intensity depends on (x, y, z), angle (θ, ϕ) and frequency (ν), making it a 6D problem and the non-local nature makes it harder to parallelize. The non-LTE 3D codes have issues with realistic 3D simulations of the chromosphere, due to large gradients in temperature and velocity. So there is a great interest to implement faster and better methods for solving non-LTE radiative transfer problems in 3D. Having a fast and efficient diagnostics code, could be important for understanding more of the complex atmospheric processes of the Sun, which require extreme resolution to study.

The non-linear multigrid scheme has shown promising results to solve these type problem with a high convergence speed-up and scalable domains. One of the first implementations of applying linear multigrid to radiative transfer was done by Steiner [11]. It was applied on a homogeneous slab of two-level atoms, showing that multigrid improved the convergence speed significantly. This was done on pure experimental basis, to explore if a multigrid method could be applied within radiative transfer. Later non-linear multigrid was implemented for solving non-LTE multilevel radiative transfer in 1D, 2D and 3D by Fabiani Bendicho et al. [1]. They used a non-linear iterative scheme called MUGA, which is based on Gauss-Seidel iterations, and with good smoothing capabilities.

In Section 4.1 we explain about the radiative transfer code, RH, then in Section 4.2 we discuss the implementation of the non-linear multigrid scheme and define some quantities to use as a convergence measure. Finally Section 4.4 shows the performance: how well the non-linear multigrid worked.

4.1. About the RH code

RH is a radiative transfer code developed by Han Uitenbroek that solves the following Eq. 2.12, 2.13 and 2.14 in 1D plane-parallel geometry/spherically symmetric geometry and 2D/3D Cartesian geometry. It is based on the MALI (Multi-

level Approximate Lambda Iteration) procedure developed by Rybicki and Hummer [9] and Rybicki and Hummer [10]. The method is based on Jacobi iterations, and this is an advantage if parallelization is needed. This formalism allows overlap of radiative transitions, and accounts for the non-linearities introduced. MALI uses preconditioning of the rate equation, which means it is using the previous iteration to arrive at a linear set of equations for the population numbers, n, which then can be solved for.

4.2. Implementation

The aim of this section is to describe the implementation of the non-linear multigrid to solve radiative transfer problems combined with an existing radiative transfer code.

4.2.1. Algorithm

RH is our iterative smoothing operator. The advantage of using an existing code, is that it has been tested and we can focus more on the implementation of the non-linear multigrid. The framework was implemented recursively as:

```
CALL V_CYCLE(level(i))
    IF level(i) == last_level THEN BEGIN

        #COARSEST LEVEL ->
        CALL RH(n.level(i),rhs.level,ndep.level(i),itr)
        #Calculate the error
        error[level(i+1)] = n_updated.level(i) - n.level(i)
ENDIF ELSE

        #GOING DOWN ->
        CALL RH(n.level(i),rhs.level(i),ndep.level(i),itr)
        n.level(i-1) = RESTRICT(n.level(i)
        rhs.level(i-1) = RESTRICT(n.level(i)
        CALL V_CYCLE(level(i-1))
    ENDELSE

        #GOING UP ->
        #UPDATE THE APPROXIMATION
        n.level(i) = INTERPOLATE(n.level(i),error(i-1))
        CALL RH(n.level(i),rhs.level(i),ndep.level(i),itr)
        error[level(i+1)] = n_updated.level(i) - n.level(i)
END
```

4.2.2. Equation system

The problem we will be solving for radiative transfer is set up as

$$\mathcal{R}^h(\mathbf{n}^h) = \mathbf{f}^h \tag{4.1}$$

where $\mathcal{R}$ is the rate equations, Eq. 2.12 and the right hand side, $\mathbf{f}$ is defined as

$$\mathbf{f}^h = \begin{bmatrix} n^l_{tot} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.2}$$

Here we have inserted the particle conservation at the ground state, which we will keep fixed through out the multigrid cycle, more explained in Section 4.2.3. For the initial starting approximation, the radiation field was set to zero in the statistical equilibrium equations for the level populations in all runs.

4.2.3. Particle Conservation

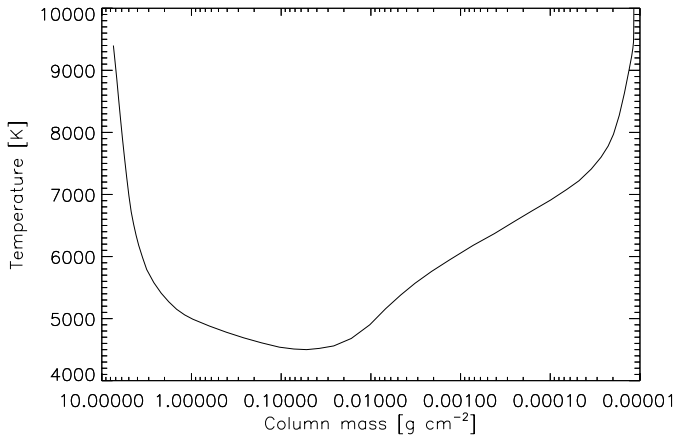
In the radiative transfer code we need to close the set of statistical equilibrium equations with the particle conservation equation. RH can set it to a pre-defined atomic level population or the level with highest population density. In the FAS algorithm we use, the same equation system is used when it is restricted to a coarser levels, but with a modified right-hand-side. To keep the system consistent trough the V-cycle, the ground level rate equation is replaced with particle conservation.

4.2.4. Atom

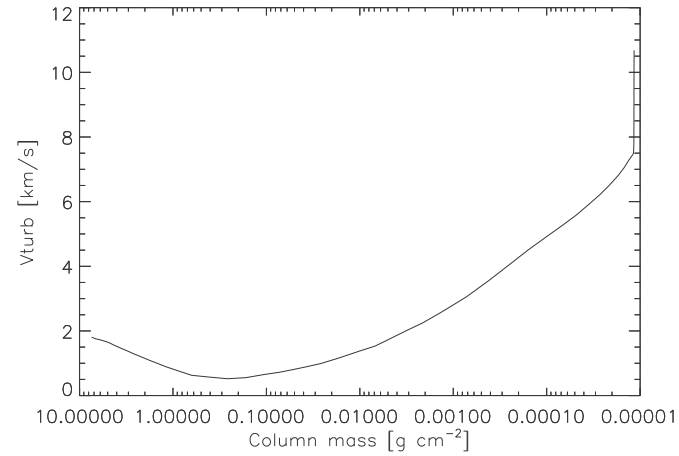
We choose two atom models,a four-level Calcium II (CaII) atom and six-level Hydrogen (H). All the lines of the atoms are done in complete frequency redistribution (CRD). We expect a larger speed-up for the Hydrogen atom when using multigrid, because of the low photon destruction probability at high optical depth in the Lyman-α line.

4.2.5. Atmosphere

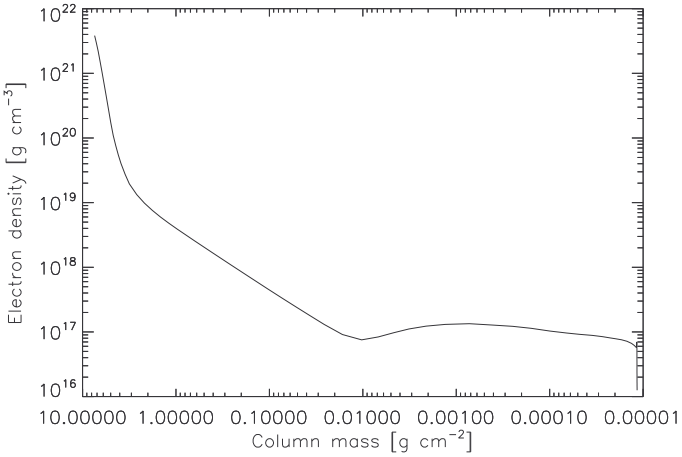
We used two atmospheres to test the multigrid method, FALC [4] and a column from a 3-D MHD atmosphere computed with the Bifrost code [5] (cb24bih). FALC is a description of the photosphere and chromosphere, which is derived empirically. Various quantities of the model atmosphere can be seen in Figure



(a) Temperature

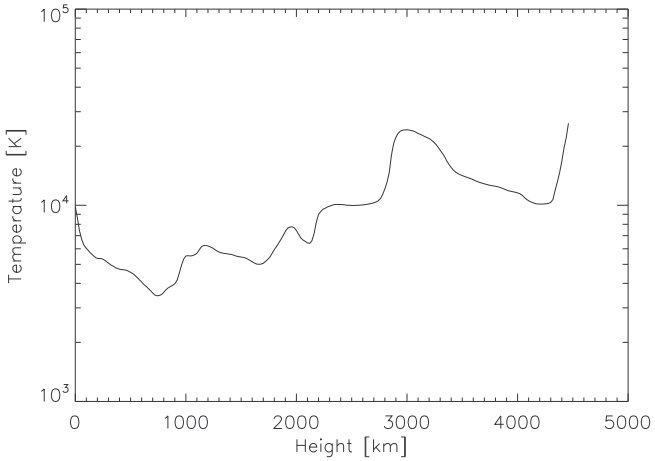


(b) Microturbulent velocity, v_t

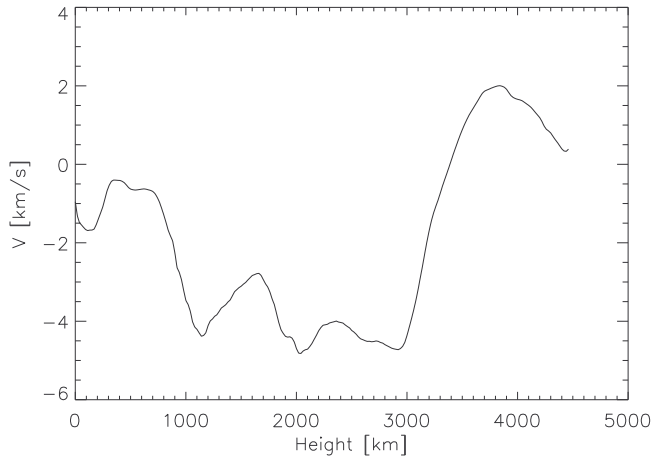


(c) Electron density, n_e

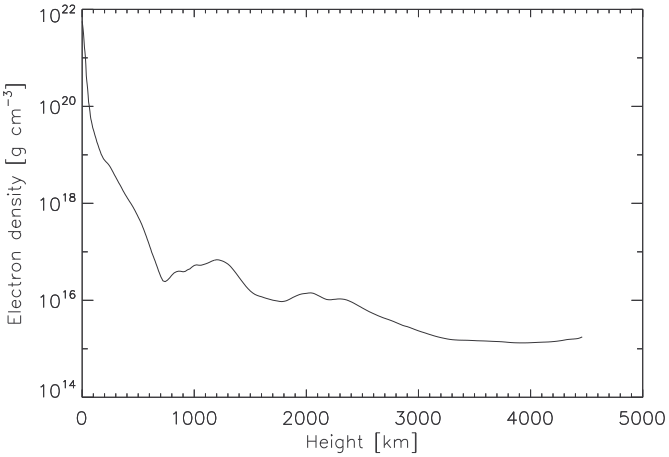
Figure 4.1.: FALC atmosphere



(a) Temperature



(b) Vertically velocity



(c) Electron density

Figure 4.2.: Bifrost atmosphere

4.1. It can be seen that the atmosphere is quite smooth. The Bifrost atmosphere, shown in Figure 4.2, is more complicated.

We used a linear interpolation to transfer atmospheres to each of the grids needed in the multigrid scheme. The highest number of grid points in the atmosphere was 1024 grid points, and the coarsest grid consists of 64 grid points.

4.3. Analyzing

4.3.1. Error

In order to study the convergence properties of the implementation of multigrid, we need to define a quantity for the error of the system. The following definition is based on Fabiani Bendicho et al. [1].

The full error for the current population number after a given number of iterations, *itr*, is given by:

$$\mathbf{e}(itr) = \mathbf{n}_l(itr) - \mathbf{n}_l\,(itr = \infty) \tag{4.3}$$

The $\mathbf{n}_l\,(itr = \infty)$, is the true solution for the population number. We solved each problem with RH without multigrid, until the relative change in populations was less then 10^{-8}. We took that solution as the true solution.

We define a quantity called the convergence error, which gives the maximum relative error at a given iteration as:

$$C_e(itr) = \left|\left|\frac{\mathbf{e}_l}{\mathbf{n}_l(itr)}\right|\right|_\infty \tag{4.4}$$

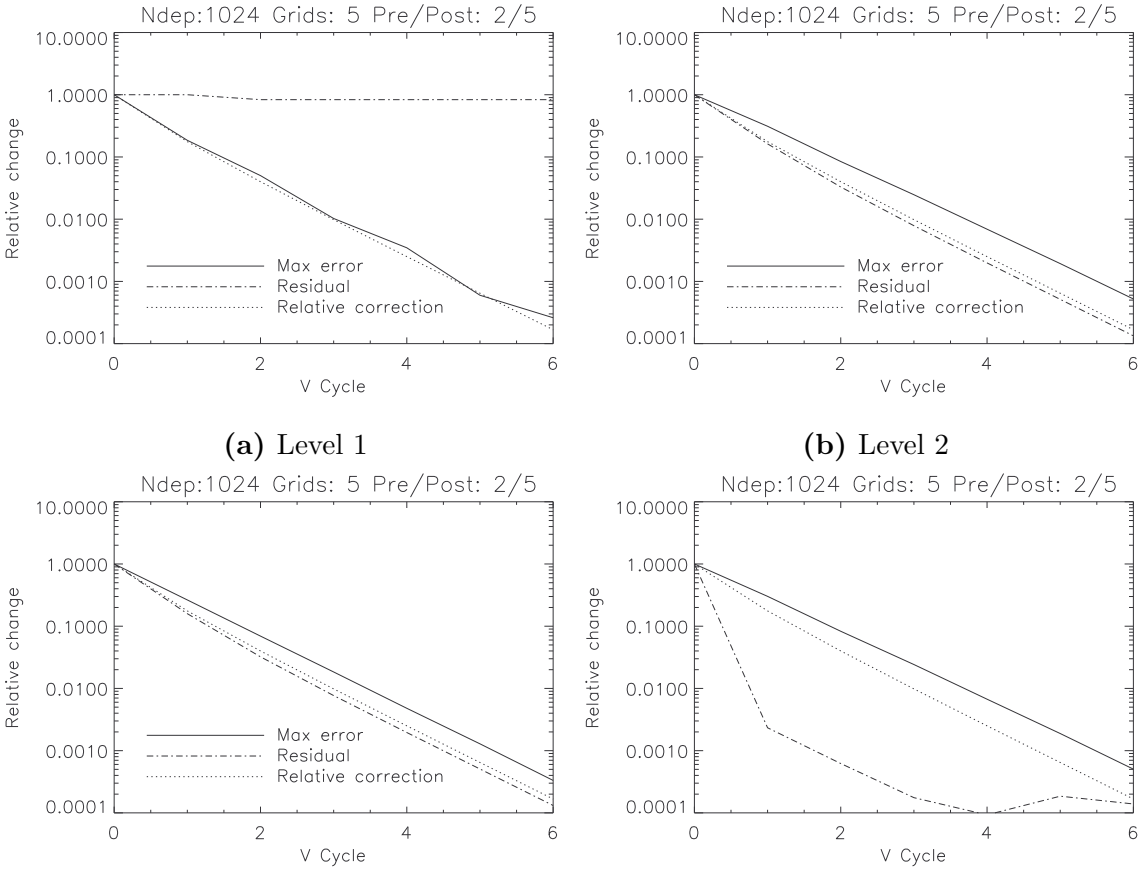
Here we applied the ∞-norm, which is defined in Eq. A.1, being the absolute max error.

4.3.2. Convergence

Doing a comparison with multigrid cycle and pure MALI iterations, it is essential to define how much computational work an iteration is on a coarser grid to compare it. Since a coarse grid, Ω^{2h}, contains half the points of the, Ω^h, we assume it takes half the computational work:

$$\#itr(\Omega^h) = \frac{1}{2}\#itr(\Omega^{2h}) \tag{4.5}$$

In this thesis we will not take into account the computational overhead of the administration part, due to how the implementation is done. The interesting part is how well multigrid performs and behaves with different setups.



(a) Level 1 **(b)** Level 2

(c) Level 3 **(d)** Level 4

Figure 4.3.: Convergence of a four-level CaII atom with FALC atmosphere as a function with V-cycles, with the properties: absolute maximum error, the residual of the equation system (Eq. 4.1) and the relative correction. All the quantities are normalized.

4.3.3. Convergence criteria

During the implementation and analysis of our multigrid method, we always had a true reference solution to compare the runs with, but in real cases this is not available. Thus, a convergence criterium is needed to stop the multigrid cycle when the desired solution is achieved. The RH code will iterate until the relative correction is smaller than a certain value. It is not certain that this method is valid for multigrid, due to the way it is solved at multiple grids together. To investigate if this holds true for the multigrid cycle, it is possible to compare the true error against some quantities we could use for deciding when the multigrid cycle has converged. One of them is the residual, $\mathbf{r} = \mathbf{f} - A\mathbf{v}$, which is a quantity that is a measure of how well the approximation fits, but it does not tell us about the magnitude of the error. So there are two different approaches to stop

the cycle: First, using the residual and second are when the change in relative populations is below a certain point, at the finest grid.

In Figure 4.3 we see the comparison with all the quantities, max error, residual and the relative correction. Since we set the particle conservation equation to ground state level, Figure 4.3a, the residual should be zero here. Due to numerical instabilities, the residual values have some small values. We take the ∞-norm of the residual, it will then never be zero.

One can see the residual and the relative correction almost behaves the same, Figure 4.3b and 4.3c, therefore one can use either quantity as a stopping criteria.

4.3.4. Post/Pre-smoothing details

How many post/pre-smoothings that are necessary mainly depends on how well the iteration scheme can smooth out the error. Which it can be mapped to a coarser grid. But for every smoothing we do, more computational work is required. Finding the optimal number for pre/post-smoothing is thus important.

These different cases have been set up to find the optimal number of the different cases with the MALI iterations scheme;

- Atmosphere, see Section 4.2.5
- Grid points
- Number of grids

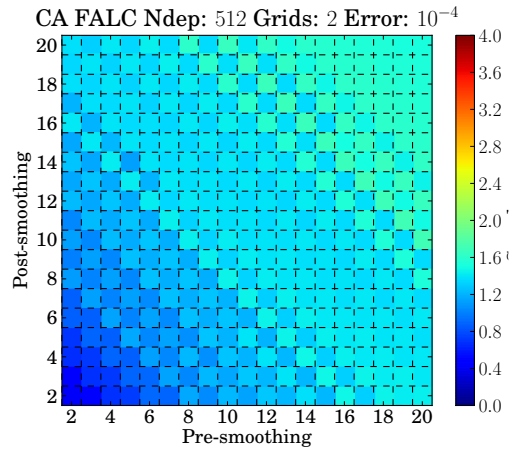
To compare the non-linear multigrid ans pure MALI iterations against each other, two criterias have been set: The first is that the non-linear multigrid has to converge in a certain number of V-cycles, if not then the computational work will exceed the pure MALI iterations, and no speed-up is gained. The second is all of the population levels have to reach a relative error below 10^{-4} at all points.

4.4. Results

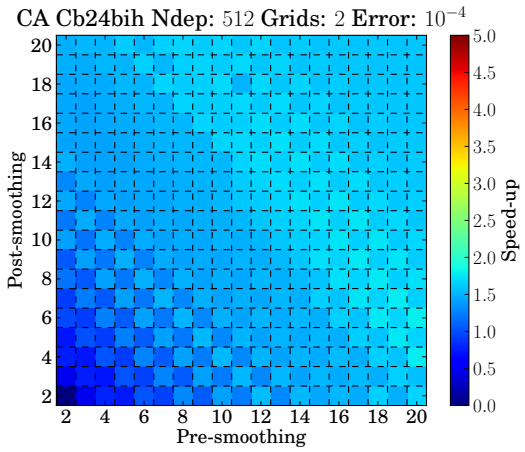
Solving the radiative transfer problem in one-dimension has been performed with a non-linear multigrid FAS scheme combined with the RH code. The simulations are done with two atoms (Sec. 4.2.4) and two atmospheres (Sec. 4.2.5).

4.4.1. Smoothing numbers

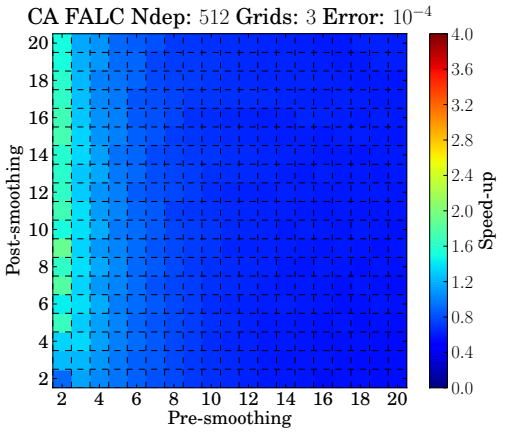
How well the post-smoothing/pre-smoothing affect the speed-up can be seen in Figures 4.4 and 4.5. Where the FALC atmosphere is in the left panel, and the Bifrost atmosphere is seen in the right panel, both of them done with the four-level CaII atom. We look at when the solution has converged to a relative maximum



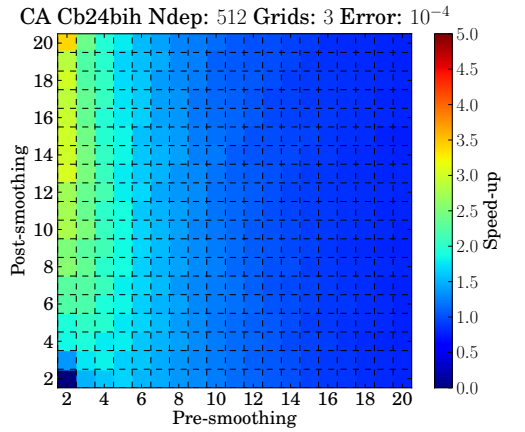
(a) FALC atmosphere with two grids



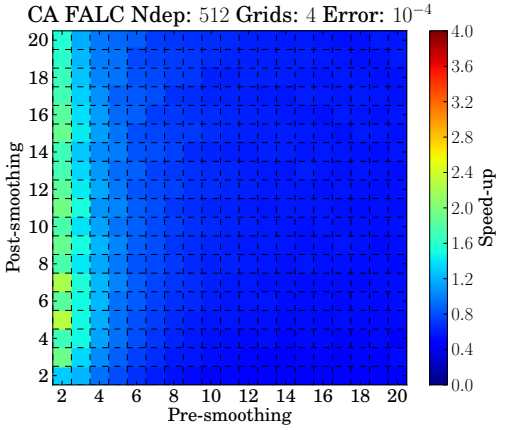
(b) Bifrost atmosphere with two grids



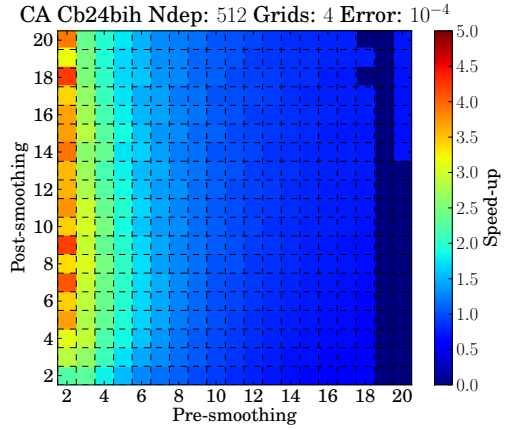
(c) FALC atmosphere with three grids



(d) Bifrost atmosphere with three grids

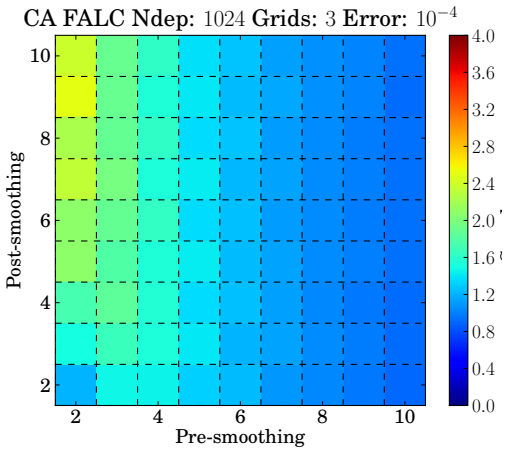


(e) FALC atmosphere with four grids

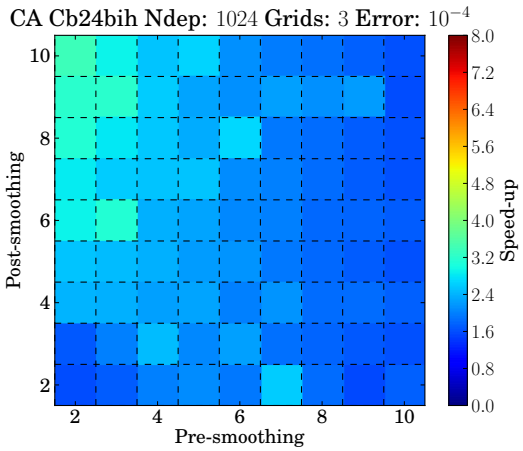


(f) Bifrost atmosphere with four grids

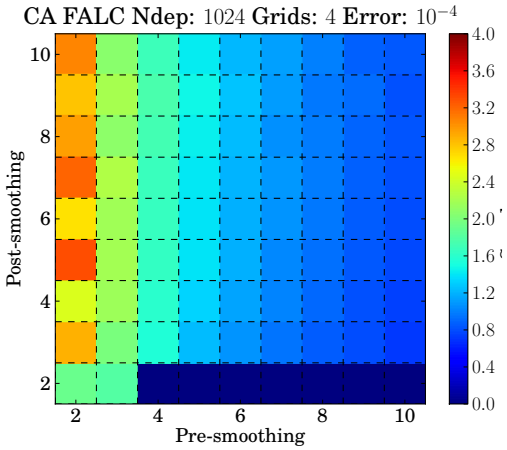
Figure 4.4.: Comparison of the speed-up gain by non-linear multigrid FAS against pure MALI iteration with 512 depth points, the left side is the FALC atmosphere and right-side is the Bifrost atmopshere. The speed-up axis goes up to four times at the left side, and five times at the right-side.



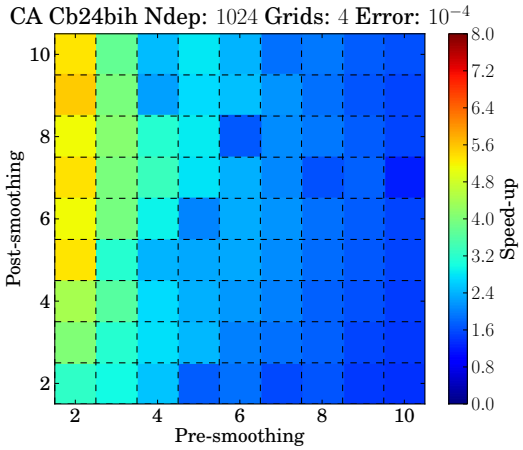
(a) FALC model with three grids



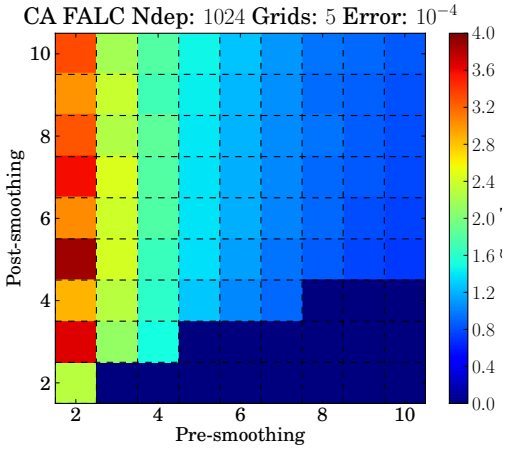
(b) Bifrost atmosphere with three grids



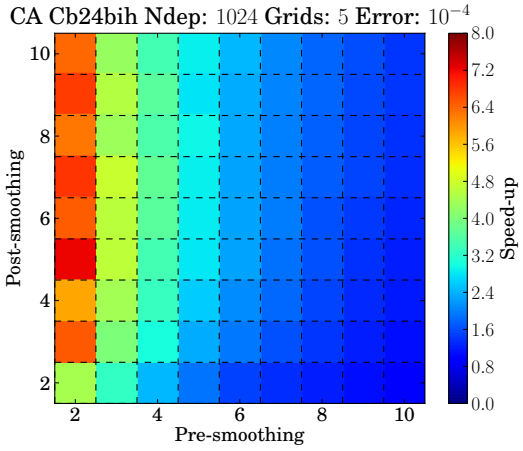
(c) FALC model with four grids



(d) Bifrost atmosphere with four grids



(e) FALC model with five grids



(f) Bifrost atmosphere with five grids

Figure 4.5.: Comparison of the speed-up gain by non-linear multigrid FAS against pure MALI iteration with 1025 depth points, the left side is the FALC atmosphere and right-side is the Bifrost atmosphere. The speed-up axis goes up to four times at the left side, and eight times at the right-side.

error 10^{-4}, comparing how well pure MALI iterations and non-linear multigrid does. The cases with just one pre-smoothing tend to diverge, therefore we will investigate where $\nu_1 \leq 2$ and $\nu_2 \leq 2$.

512 depth points

In Figure 4.4a and 4.4b with two grids, the coarsest grid consists of 256 depth points. A linear behavior can be seen with the pre/post-smoothing. At the coarsest grid the multigrid converge to the right solution for that grid, $itr \to \infty$. Since we just have two-grid in this test, the pre/post-smoothing iterations will be coupled, $\nu_1 + \nu_2$, since this is just performed on the finest grid. The speed-up for each of the atmospheres, is similar, around 2 times with optimal choosing of pre/post-smoothing.

Adding more grids we see the power of multigrid, in Figure 4.4c and 4.4d, having three grids, where the coarsest grid consist of 128 depth points. Already with 2 pre-smoothing is enough to have a smooth residual and approximation to represent it at a coarse grid, while increasing it will just increase the computational work and gain little or any more improvement over the smoothness to the residual/approximation. As a result of this, the post-smoothing defines how much speed-up we gain. In the Bifrost atmosphere the speed-up gets larger as almost linear as the post-smoothing increase, with the three grids. The highest speed-up gain is done when we have applied four grids, seen in Figure 4.4e and 4.4f. In the Bifrost atmosphere at $19-20$ pre-smoothing, we start to see that the method start to converge too slowly, and it is not worth doing more multigrid cycles.

1024 depth points

Increasing the number of depth points to 1024, seen in Figure 4.5, shows how well the non-linear multigrid works for a high number of depth points. Since multigrid scales linearly with the number of operations for solving n unknowns, it can be interpreted that the pure MALI iterations deteriorate when increasing the number of grid points. Here we can clearly see the big advantage of multigrid, the convergence speed is independent of the grid spacing.

In Figure 4.5a and Figure 4.5b, there is no clear linear behavior as in the other cases when using different parameters for the pre/post-smoothing. This can especially be seen in the Bifrost atmosphere. Still it gives a high convergence, for the FALC 2.5 times speed-up, and the Bifrost atmosphere 3.8 times speed-up.

Adding another grid, Figure 4.5c and 4.5d , we start to see more linear behavior in both atmospheres, indicating that two pre-smoothing also holds here. In the FALC atmosphere, we start to see higher convergence for the smooth atmosphere, reaching above 3 times the speed-up. And in Bifrost it have reached above 5 times the speed-up.

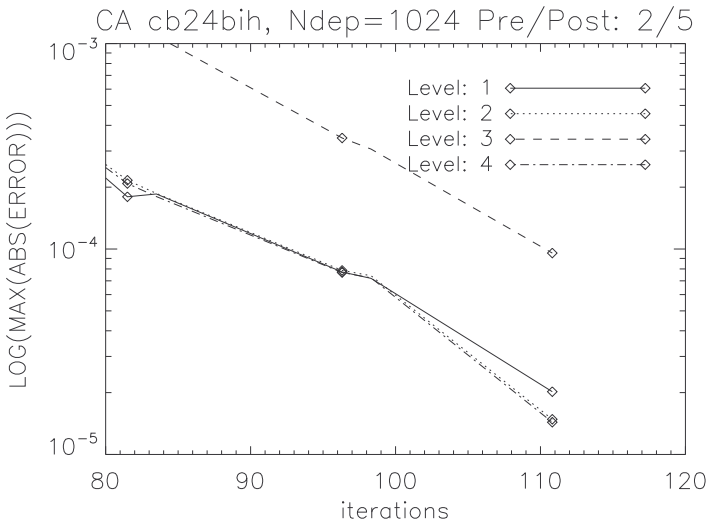


Figure 4.6.: The relative max error for the population number as a function of iterations, for each level which is denoted with different lines, where the multigrid scheme stops when all of them is below the maximum relative error 10^{-4}. The diamond symbol denote post/pre-smoothing.

At five-grid, Figure 4.5e and 4.5f, we see that the convergence speed is high in both atmospheres. In the FALC atmosphere we get 4 times the speed-up, with 2 pre-smoothing and 5 post-smoothing, while with the Bifrost atmosphere we get 8 times the speed-up, with 2 pre-smoothing and 5 post-smoothing. Remember at the coarsest grid now, the computational cost is less than $\frac{1}{2^{n_g}-1}$, where the n_g is the number of grids. So essentially, the cost of doing one iteration at the $n_g = 5$ compared to one iterations at the finest grid, is a factor of 1/16 smaller. Adding more grids we reduce the computational cost and effective convergence faster.

Based on these results, we find that choosing two pre-smoothing and five post-smoothing leads to the optimal convergence speed in the V-cycle for non-linear multigrid.

Oscillating behavior in convergence speed-up

In Figure 4.5e and 4.5f, we can see an oscillating behavior in the convergence speed-up for the post-smoothing numbers. When the post-smoothing is 2, we have a convergence speed at 3.5 speed-up, while at a post-smoothing 3, it falls down to a 2.8 speed-up, hence oscillating.

One factor that contribute to this effect is due to where the V-cycle ends after the post-smoothing. Since the criteria to stop multigrid cycle is when the relative max error is less than 10^{-4} for every population level, it is not certain that all the levels reach this criteria at the same V-cycle. As seen in Figure 4.6, level 3 has reached the criteria one V-cycle later compared to the other levels, which already

had achieved the wanted maximum relative error. Since the cost of a V-cycle is determined of how many pre/post-smoothing and coarse-grid-correction done, and how many grids applied, the computational cost is much more than a one time iteration with a MALI. To reach the criteria one more V-cycle is needed. Then the convergence speed goes down, since the computational cost is high.

This factor alone can not contribute to the oscillating behavior. But there was not sufficient time to investigate this problem closer.

4.4.2. Number of coarse grids

The convergence rate is influenced by how many grids are applied. In Figure 4.4 and 4.5, we can clearly see that increasing the number of grids increases our convergence speed. This follows from the fact that the computational cost decreases by a factor of $\frac{1}{2}$ for each grid we apply, and removes the low-frequency component faster.

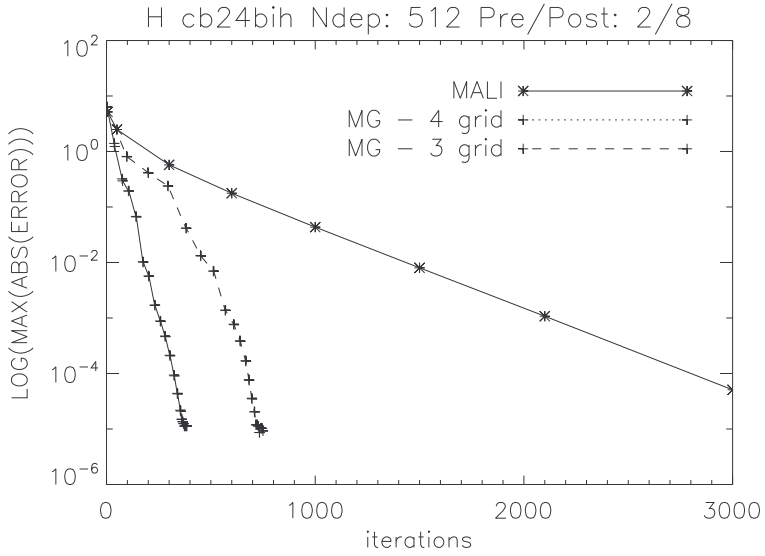
4.4.3. Hydrogen

The maximum relative error in the population numbers as a function of iteration number is illustrated in Figure 4.7 for a Hydrogen six-level atom with multigrid cycle and pure MALI iterations. This is tested in an atmosphere with 512 depth points. For the FALC atmosphere, which can be seen from Figure 4.7b, the convergence for both cases for the three and four grid is rapid compared with pure MALI iterations. There is not so much improvement of using a four-grid over the three grid, which was also seen in the Ca II case too. The Bifrost atmosphere, seen in Figure 4.7a, has much higher convergence speed-up in both cases, and it depends on how many grids are used. This was also seen in the Ca II case, with the Bifrost atmosphere.

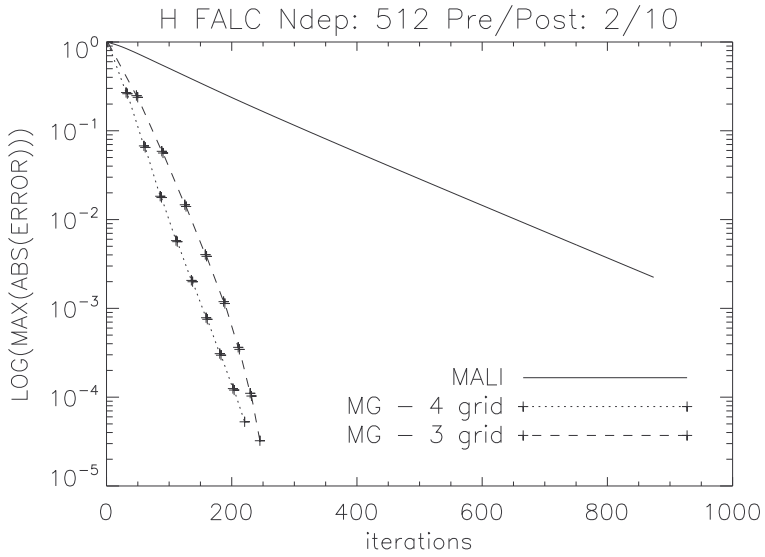
4.4.4. Restriction

Injection and full-weighting has been tested if it affects the convergence of the FAS algorithm. Since it is possible to use different operators on the restriction on the population and the residual, this also has been tested. In Figure 4.8 we have applied injection and full-weighting on the population and residual. Where it stalls, is where the FAS V-cycle does not converge any further. In Figure 4.8a we see that the full-weighting has a small advantage, but stalls at a slightly higher maximum error. While in Figure 4.8b, it converge at the same rate, and have almost no significant change.

Based on several experiments, we find that there is no significant difference while using the different operators in this case, both on the operator for restrict the residual and population.

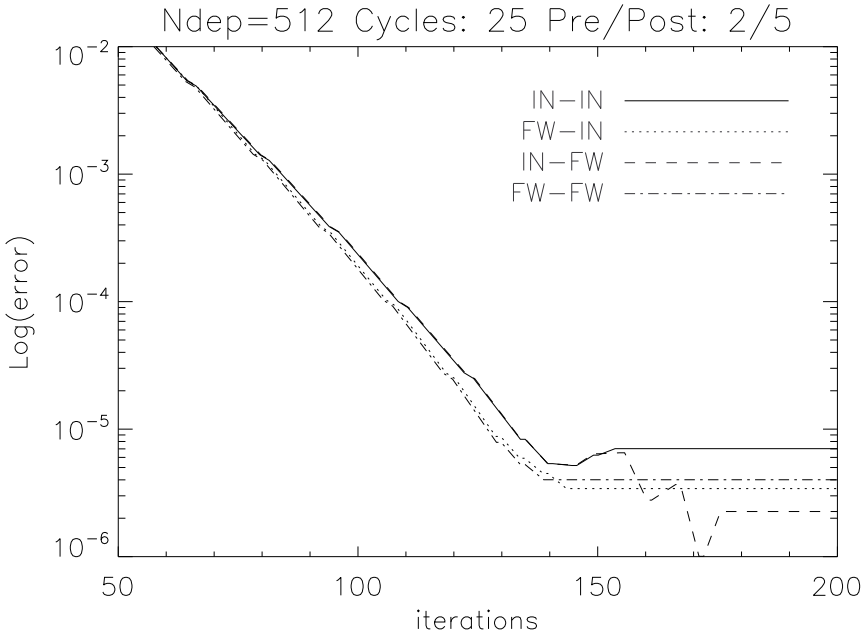


(a) Bifrost atmosphere

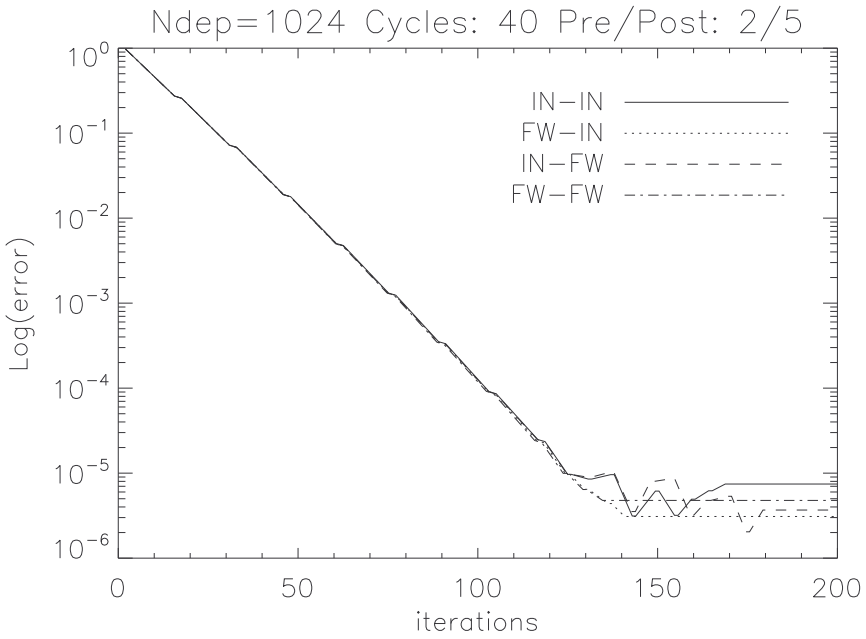


(b) FALC atmosphere

Figure 4.7.: The convergence error for a six-level Hydrogen atom with with two atmospheres versus iteration, one MALI iteration in the finest grid. The solid line is pure MALI iterations and the dashed line is a three grid multigrid, while dotted line is a four grid multigrid.



(a) FALC atmosphere with four grids, consists of 512 depth points



(b) FALC atmosphere with five grids, consists of 1024 depth points

Figure 4.8.: The maximum relative error in the population numbers as a function of iteration number, comparison with different restriction operator, injection (*in*) and full-weighting (*fw*), for the residual, **r**, and population, **n**. The solid lines refer to *in* operator on both, dotted lines to *fw* on **r** and *in* on population, dashed-lines to *in* on **r** and *fw* on **n**, while the dotted dashed-lines to *fw* on both.

5. Conclusion and outlook

A framework around the RH code has been implemented with a non-linear multigrid scheme, providing a tool for further investigating possibilities with multigrid. This is one of the approaches to solve the convergence issues with a non-LTE 3D code, which is needed to investigate more demanding problems, requiring more computing time and memory.

The non-linear multigrid scheme combined with the RH code has been tested and it is found to provide results consist with earlier work. It has been show that the scheme is comparable with the results done by Fabiani Bendicho et al. [1], which proves that MALI also can perform high convergence. Our optimal numbers of pre/post-smoothing are also consistent with Štěpán and Trujillo Bueno [13].

5.1. Improvements

Here are some shortcomings that could be interesting to further investigate with a multigrid scheme.

Adaptive grids/non-uniform grid

In the implementation of multigrid we used uniform grids, to study more complicated atmospheres it may be needed to implement adaptive grids. This can take in account the very large gradients that appear in temperature and velocity in the atmospheres without introducing a fine-grid in the whole domain. One of the methods is called "Fast Adaptive composite", which is described in "A Multigrid Tutorial, Second Edition" [14].

Ng-acceleration

There exists accelerations methods, such as the Ng-acceleration, which could be utilized with multigrid, especially at the coarsest grid. This could reduce our computational cost, if few grids are applied.

Appendices

A. Definition

There are two norms being used throughout this thesis which is to analyze the error. Since the error is also a vector, it may be measured with any standard vector norms,

Definition of them are listed here:

The maximum norm

$$||e||_\infty = \max_{1 \le i \le n} |e_i| \tag{A.1}$$

Bibliography

[1] P. Fabiani Bendicho, J. Trujillo Bueno, and L. Auer. Multidimensional radiative transfer with multilevel atoms. II. The non-linear multigrid method. 324:161–176, August 1997.

[2] R.P. Fedorenko. A relaxation method for solving elliptic difference equations. {*USSR*} *Computational Mathematics and Mathematical Physics*, 1(4):1092 – 1096, 1962. ISSN 0041-5553.

[3] R.P. Fedorenko. The speed of convergence of one iterative process. {*USSR*} *Computational Mathematics and Mathematical Physics*, 4(3):227 – 235, 1964. ISSN 0041-5553.

[4] J. M. Fontenla, E. H. Avrett, and R. Loeser. Energy balance in the solar transition region. III - Helium emission in hydrostatic, constant-abundance models with diffusion. *apj*, 406:319–345, March 1993. doi: 10.1086/172443.

[5] B. V. Gudiksen, M. Carlsson, V. H. Hansteen, W. Hayek, J. Leenaarts, and J. Martínez-Sykora. The stellar atmosphere simulation code Bifrost. Code description and validation. *aap*, 531:A154, July 2011. doi: 10.1051/0004-6361/201116520.

[6] Gundolf Haase and Ulrich Langer. Multigrid methods: from geometrical to algebraic versions. In Anne Bourlioux, MartinJ. Gander, and Gert Sabidussi, editors, *Modern Methods in Scientific Computing and Applications*, volume 75 of *NATO Science Series*, pages 103–153. Springer Netherlands, 2002. ISBN 978-1-4020-0782-8. doi: 10.1007/978-94-010-0510-4_4. URL `http://dx.doi.org/10.1007/978-94-010-0510-4_4`.

[7] Wolfgang Hackbush. *Multi-Grid Methods and Applications*. Springer-Verlag Berlin, Hedelberg, 1985.

[8] R. J. Rutten. *The Generation And Transport of Radiation*. Utrecht, fifth edition, 1988.

[9] G. B. Rybicki and D. G. Hummer. An accelerated lambda iteration method for multilevel radiative transfer. I - Non-overlapping lines with background continuum. 245:171–181, May 1991.

[10] G. B. Rybicki and D. G. Hummer. An accelerated lambda iteration method for multilevel radiative transfer. II - Overlapping transitions with full continuum. 262:209–215, August 1992.

[11] O. Steiner. Fast solution of radiative transfer problems using a method of multiple grids. 242:290–300, February 1991.

[12] Michael Stix. *The Sun: An Introduction*. Springer, 2002.

[13] J. Štěpán and J. Trujillo Bueno. PORTA: A three-dimensional multilevel radiative transfer code for modeling the intensity and polarization of spectral lines with massively parallel computers. *aap*, 557:A143, September 2013. doi: 10.1051/0004-6361/201321742.

[14] Steven F. McCormick William L. Briggs, Van Emden Henson. *A Multigrid Tutorial, Second Edition*. Siam, 2000.